Structured Programming Language (CSE113)
Lecture #05
Switch Case in C programming

Switch statement

- A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each switch **case**.
- Switch statement is used to solve multiple option type problems for menu like program, where one value is associated with each option.
- Switch case statements mostly used when we have number of options (or choices) and we may need to perform a different task for each choice.

Syntax of switch...case

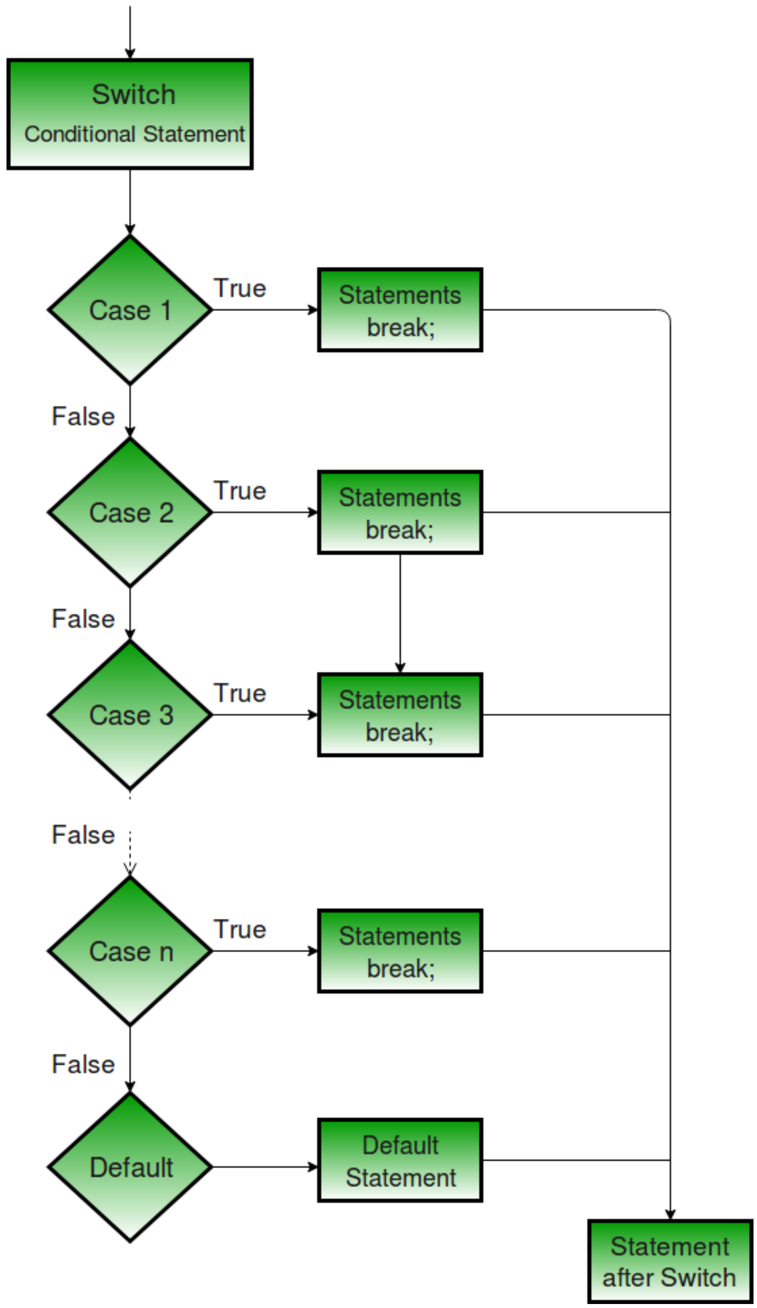
```
switch(n)
{
    case constant1:
        // code to be executed if n is equal to constant1;
        break;

    case constant2:
        // code to be executed if n is equal to constant2;
        break;
        .
        .
        .
    default:
        // code to be executed if n doesn't match any constant
}
```

Break statement

Break statements are useful when you want your program-flow to come out of the switch body. Whenever a **break** statement is encountered in the switch body, the execution flow would directly come out of the switch.

switch Statement Flowchart



Few points about Switch Case

1) Case doesn't always need to have order 1, 2, 3 and so on. It can have any integer value after case keyword. Also, case doesn't need to be in an ascending order always, you can specify them in any order as per the need of the program.
2) You can also use characters in switch case. for example –

```
int main()
{
     char ch='b';
     switch (ch)
     {
         case 'd':
            printf("Case D ");
            break;
         case 'b':
            printf("Case B");
            break;
         case 'c':
            printf("Case C");
            break;
         case 'z':
            printf("Case Z ");
            break;
         default:
            printf("Default ");
    }
    return 0;
}
```

Output:

```
Case B
```

3) The expression provided in the switch should result in a constant value otherwise it would not be valid.

Valid expressions for switch –

// Constant expressions allowed

switch(1+2+23)

switch(1*2+3%4)

// Variable expression are allowed
// they are assigned with fixed values

switch(a*b+c*d)

switch(a+b+c)

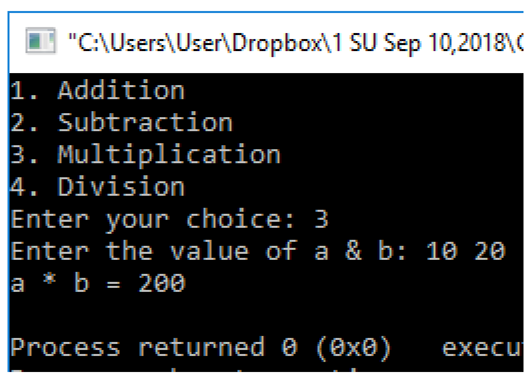
4) Nesting of switch statements are allowed, which means you can have switch statements inside another switch. However nested switch statements should be avoided as it makes program more complex and less readable.

Q1. Write a C Program to perform the following operations between two integers using switch case:

1. Addition
2. Subtraction
3. Multiplication
4. Division

<u>Source Code</u>

```
#include<stdio.h>
int main()
{
   int choice;
   int a,b;
   printf("1. Addition\n2. Subtraction\n3. Multiplication\n4. Division\n");
   printf("Enter your choice: ");
   scanf("%d",&choice);
   switch(choice){
      case 1:
         printf("Enter the value of a & b: ");
         scanf("%d %d",&a,&b);
         printf("a + b = %d\n",a+b);
      break;
      case 2:
         printf("Enter the value of a & b: ");
         scanf("%d %d",&a,&b);
         printf("a - b = %d\n",a-b);
      break;
      case 3:
         printf("Enter the value of a & b: ");
         scanf("%d %d",&a,&b);
         printf("a * b = %d\n",a*b);
      break;
      case 4:
         printf("Enter the value of a & b: ");
         scanf("%d %d",&a,&b);
         printf("a / b = %d\n",a/b);
      break;
      default:
      printf("Invalid choice\n");
   }
   return 0;
}
```



Exercise Questions

1. Write a C program to read weekday number (1-7) and print weekday name (Saturday, Sunday …) using switch.
2. Write a C program to read gender (M/F) and print corresponding gender using switch.
3. Write C program to check whether a character is VOWEL or CONSONANT using switch.
4. Write a C program to check whether number is EVEN or ODD using switch.
5. Write a C program to find number of days in a month using switch case.
6. Write a C program to design calculator with basic operations using switch.